Variational Density Propagation Continual Learning

Christopher F. Angelini
Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, New Jersey, USA
angelinic0@rowan.edu

Nidhal C. Bouaynaya
Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, New Jersey, USA
bouaynaya@rowan.edu

Ghulam Rasool
Depts. of Machine Learning and Neuro-Oncology
Moffitt Cancer Center & Research Institute
Tampa, Florida, USA
Ghulam.Rasool@moffitt.org

2023 International Symposium on Image and Signal Processing and Analysis (ISPA) | 979-8-3503-1536-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISPA58351.2023.10279186

Abstract—Deep Neural Networks (DNNs) deployed to the real world are regularly subject to out-of-distribution (OoD) data, various types of noise, and shifting conceptual objectives. This paper proposes a framework for adapting to data distribution drift modeled by benchmark Continual Learning datasets. We develop and evaluate a method of Continual Learning that leverages uncertainty quantification from Bayesian Inference to mitigate catastrophic forgetting. We expand on previous approaches by removing the need for Monte Carlo sampling of the model weights to sample the predictive distribution. We optimize a closed-form Evidence Lower Bound (ELBO) objective approximating the predictive distribution by propagating the first two moments of a distribution, i.e. mean and covariance, through all network layers. Catastrophic forgetting is mitigated by using the closed-form ELBO to approximate the Minimum Description Length (MDL) Principle, inherently penalizing changes in the model likelihood by minimizing the KL Divergence between the variational posterior for the current task and the previous task's variational posterior acting as the prior. Leveraging the approximation of the MDL principle, we aim to initially learn a sparse variational posterior and then minimize additional model complexity learned for subsequent tasks. Our approach is evaluated for the task incremental learning scenario using density propagated versions of fully-connected and convolutional neural networks across multiple sequential benchmark datasets with varying task sequence lengths. Ultimately, this procedure produces a minimally complex network over a series of tasks mitigating catastrophic forgetting.

Index Terms—Continual Learning, Bayesian Deep Learning, Deep Variational Inference, Minimum Description Length Principle, Density Propagation

I. Introduction

A commonly held assumption in deep learning is a network's training and test data distributions are static and accurately represent its deployed environment. However, Deep Neural Networks (DNNs) deployed in real-world environments are regularly subject to out-of-distribution (OoD) data, various types of noise, and shifting conceptual objectives [1]. Input patterns not generalized by the network after training may inadvertently trigger features in the network leading to incorrect results with high confidence during deployment.

This work was supported by the National Science Foundation Awards NSF ECCS-1903466 and NSF OAC- 2234836. We are also grateful to UK EPSRC support through EP/T013265/1 project NSF-EPSRC: ShiRAS. Towards Safe and Reliable Autonomy in Sensor Driven Systems.

While adapting to data drift can be achieved by retraining a DNN on an entire dataset comprised of the original samples and new drifted samples, there are situations where this may be infeasible or unreasonable due to time, cost, computing, or retained data constraints. Additionally, simply training on the new information from the data drift will result in a phenomenon called catastrophic forgetting, causing representations of newly trained information to interfere with or overwrite previous representations, resulting in a drop in performance on previously trained samples [2], [3].

Reasonable stand-alone performance in real-world environments requires DNNs to indicate OoD data samples and adapt to these anonymous samples without sacrificing performance on previously trained information. To achieve these requirements, the sequential nature of Bayesian inference naturally lends itself as a mathematical framework to quantify uncertainty and continually adapt to new data without forgetting previous information [4]. Variational methods using the Evidence Lower Bound (ELBO) to approximate the true parameter posterior in Bayesian inference are asymptotically equivalent to the Minimum Description Length principle as the sample size grows to infinity [5]. Both approaches regularize optimization by penalizing model complexity, preventing the overfitting of training data. We propose to leverage this concept for Continual Learning (CL) to mitigate catastrophic forgetting by minimizing changes to the network parameters over subsequent tasks.

While many deep learning variational approximations of Bayesian Inference exist, such as Bayes-by-Backprop (BBB) [6], Monte-Carlo Dropout (MCDrop) [7], and have been applied to Continual Learning [8], most rely on Monte Carlo sampling during training and testing to estimate predictive uncertainty. Uncertainty learned by these methods is only based on a few samples of the variational posterior requiring the reparameterization trick [6], [9] using "noise variables" to decouple the network parameters from the variational distribution for dealing with non-differentiable and/or complex models. A major limitation of these approaches is the significant time and computational cost imposed by sampling the variational posterior, where more samples must be drawn to better estimate the uncertainty about a prediction.

Instead, we propose using the general, model-agnostic

framework, Variational Density Propagation (VDP) demonstrated in our previous works [10], [11], to quantify the predictive uncertainty and aid in the mitigation of catastrophic forgetting via model complexity regularization. This process removes the need for Monte Carlo sampling of the variational posterior by propagating the first two moments of the variational distribution, i.e. mean and covariance, using a first-order Taylor series approximation. To enhance variance expression in the Variational Density Propagation framework, we impose independence across all network parameters resulting in each random variable parameter having a unique mean and variance. We approximate the propagation of the covariance matrix through each layer as a vectorized diagonal of the full covariance matrix representing the variance of each propagated feature to aid computational efficiency.

By modeling changes in each parameter's variational posteriors as a change in model complexity, changes in parameters can be penalized via the approximation of the Minimum Description Length Principle. This approximation is achieved by minimizing the Kullback-Leibler (KL) divergence between the variational posterior and the network prior inherent to the Evidence Lower Bound objective of Variational Inference. While learning additional information and overwriting already trained representations may not strictly increase model complexity, setting the network prior to the variational posterior from the previously learned task will construe any change to the previous variational posterior as increased model complexity. In this manner, an additional task or a data drift's representation can be learned with minimal changes to the variational posterior from the previous task, preserving representations and mitigating catastrophic forgetting.

To demonstrate this approach, our contributions are as follows:

- We develop a fully factorized version of the Variational Density Propagation framework, which uses Taylor-series approximation to propagate the first two moments of the variational distribution, with an approximation of the propagated covariance matrix for reduced computational requirements.
- We convert the approximation of Monte Carlo sampling to the propagation of variational moments to approximate the Minimum Description Length Principle.
- We apply our framework to the problem of task incremental learning by imposing a model complexity cost over a series of tasks.
- We demonstrate catastrophic forgetting mitigation over task incremental learning with multiple sequential benchmark datasets and compare our results to Monte Carlo sampling-based approaches and baseline performance metrics in the Bayesian and deterministic settings.

II. Bayesian Deep Learning

In Deep Bayesian Inference, network parameters, W, are represented as random variables with some prior distribution $W \sim p(W)$. After observing some training set $D = \{X^{(i)}, y^{(i)}\}_{i=1}^{n}$, Bayes' Rule is used to determine the posterior distribution $p(W|D)$. The predictive distribution, $p(y|X, D)$ is determined by marginalizing the parameters, W, within the posterior distribution. Marginalization can also be performed previously unseen data $\tilde{X}$ with corresponding output $\tilde{y}$ to perform inference after training, as shown in Equation 1.

$$p(\tilde{y}|\tilde{X}, D) = \int p(\tilde{y}|\tilde{X}, W)p(W|D) \quad dW \tag{1}$$

The mean of the predictive distribution represents the network's prediction. The predictive variance represents the statistical uncertainty of the network attached to the same prediction. The predictive variance is construed as confidence in an estimation.

A. Variational Inference

Despite having an analytical formulation, applying Bayesian Inference to DNNs is intractable due to the required integration of all network parameters [6]. To avoid this issue, Variational Inference (VI) is a common technique used to estimate the posterior distribution of a network by converting the intractable, analytical problem of solving the posterior distribution to an optimization problem. This approach imposes a distribution of variational parameters, $q_\theta(\boldsymbol{\Omega})$, over the network parameters and minimizes the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior distribution, shown in Equation 2.

$$\begin{aligned} \min_\theta KL[q_\theta(\Omega)||p(\Omega|D] &= \min_\theta \int q_\theta(\Omega) \ln \frac{q_\theta(\Omega)}{p(\Omega|D)} d\Omega \\ \min_\theta KL[q_\theta(\Omega)||p(\Omega|D)] &= \min_\theta E_{q_\theta(\Omega)} \ln \frac{q_\theta(\Omega)}{p(\Omega|D)} \end{aligned} \tag{2}$$

However, directly minimizing the KL divergence between the variational parameters and the true posterior is also intractable as determining the true posterior from Bayes' rule still requires the evidence derived from the integration over the product of the likelihood function and the prior distribution $p(D) = \int p(D|\Omega)p(\Omega)d\Omega$. Instead, VI maximizes an equivalent quantity called the Evidence Lower Bound (ELBO) which can be derived from Equation 2 using Bayes' Rule to rewrite the true posterior, as shown in Equation 3. The log marginal likelihood of the data, $\ln p(D)$, is fixed and does not depend on the variational distribution and can be grouped with the KL divergence between the variational posterior and the true posterior becoming the lower bound on the evidence.

$$\begin{aligned} KL[q_\theta(\Omega)||p(\Omega|D)] &= E_{q_\theta(\Omega)}[\ln q_\theta(\Omega) - \ln p(\Omega|D)] \\ KL[q_\theta(\Omega)||p(\Omega|D)] &= E_{q_\theta(\Omega)} \ln q_\theta(\Omega) - \ln \frac{p(y|X,\Omega)p(\Omega)}{p(D)} \\ KL[q_\theta(\Omega)||p(\Omega|D)] &= E_{q_\theta(\Omega)} \ln \frac{q_\theta(\Omega)}{p(\Omega)} - \ln p(y|X,\Omega) + \ln p(D) \\ -ELBO &= -E_{q_\theta(\Omega)}[\ln p(y|X,\Omega)] + KL_{q_\theta(\Omega)}[q_\theta(\Omega)||p(\Omega)] \end{aligned} \tag{3}$$

The optimization objective for Variational Inference is the maximization of the ELBO as shown in Equation (4). Maximizing this quantity ultimately results in the maximization

of the log-likelihood of data given the network parameters and the minimization of the KL divergence between the variational posterior and the prior distribution.

$$\varphi^{*} = \operatorname{argmax} E_{q_\theta(\Omega)}[\ln p(y|X,\Omega)] - KL_{q_\theta(\Omega)}[q_\theta(\Omega)||p(\Omega)] \quad (4)$$

B. Moments Propagation

The goal for moments propagation is to produce the mean and the covariance of the predictive distribution from which the network's success or failure can be gauged by relating the predictive distribution's mean to the label weighted by the predictive distribution's variance. To achieve moments propagation, all network operations are replaced by operations of random variables. Each algebraic operation in a deterministic network is replaced by the multiplication of a random variable with a constant, multiplication of two random variables, or approximating the non-linear transformation over random variables using a first-order Taylor-series approximation [10]. As a result, the first and second moments of the variational distribution, mean and covariance, can be propagated layer by layer through the entire network.

For simplicity of notation and without loss of generality, we consider a fully connected model. Let $x \in R^n$ be the input to a layer with mean μ_x and covariance matrix Σ_x. We assume that the j^{th} model parameter w_j follows a Normal distribution $w_j \sim N(\mu_{w_j}, \sigma^2_{w_j})$. The model parameters are assumed to be independent from each other and independent from the input. The first-order Taylor series is used to approximate the first two moments after a non-linear activation function. In the following, we present the derivations pertaining to the various model layers.

1) Propagation through the k^{th} linear layer: To streamline notation, we will exclude the reference to k in our representation. Let $z = W^T x + b$, where, for the k^{th} layer, $W \in R^{n \times m}$ is a random matrix of weights, $b \in R^m$ is a random vector of biases, and $z \in R^m$ is the resulting random vector. Let $W = [w_1, \cdots, w_m]$, where w_i is the i^{th} column of W, with the mean and covariance of w_i represented as μ_{w_i} and Σ_{w_i}. It follows that the mean and variance elements, μ_{z_i} and $\sigma^2_{z_i}$, contained within the resulting random vector z can be derived for elements $i = 1 \cdots m$ with the matrix-vector multiplication as shown in Equation (5).

$$\begin{aligned} \mu_{z_i} &= \mu_{w_i}^T \mu_x + \mu_{b_i} \\ \sigma^2_{z_i} &= \operatorname{tr}(\Sigma_{w_i} \Sigma_x) + \mu_x^T \Sigma_{w_i} \mu_x + \mu_{w_i}^T \Sigma_x \mu_{w_i} + \sigma^2_{b_i} \end{aligned} \quad (5)$$

The input data for the first layer of the network $k = 0$ is treated as deterministic where, in Equation (5), $\Sigma_x = 0$. The resulting covariance for $\sigma^2_{z_i}$ only depends on $\mu_x{}^T \Sigma_{w_i} \mu_x$.

For computational efficiency, moment propagation is reformulated to only propagate the diagonal variance elements of the covariance information through the network. As a result of this further approximation, Equation (5) can be reformulated as Equation (6) below, where $\mu_{w_{i,h}}$ and $\sigma^2_{w_{i,h}}$ are the h^{th} element of μ_{w_i} and the h^{th} diagonal element of Σ_{w_i}, respectively.

$$\begin{aligned} \mu_{z_i} &= \mu_{w_i}^T \mu_x + \mu_{b_i} \\ \sigma^2_{z_i} &= \sum_{h=1}^{n} (\sigma^2_{x_h} \sigma^2_{w_{i,h}} + \mu^2_{x_h} \sigma^2_{w_{i,h}} + \sigma^2_{x_h} \mu^2_{w_{i,h}}) + \sigma^2_{b_i} \end{aligned} \quad (6)$$

It is important to note that as Equation (6) is a further approximation, there is a loss in the fidelity of the propagated variances due to subsequent network layers' reliance on the covariance elements of previous layers to compute the true covariance values. Convolution operations are treated in the same manner where kernels and underlying image patches are vectorized prior to moment propagation.

2) Propagation through a non-linear activation: Let $g = \Psi(z)$ represent some non-linear activation function (e.g. ReLU, Hyperbolic Tangent, Softmax) of a random vector input $z \in R^m$ with mean μ_z and covariance Σ_z. The mean and covariance for the resulting random vector g can be approximated using the first-order Taylor series approximation [10]–[12] in Equation (7) where $\odot$ is the element-wise product of the incoming covariance matrix, Σ_z, and the squared gradient, ∇, of non-linear function with respect to the incoming mean, μ_z.

$$\begin{aligned} \mu_g &\approx \Psi(\mu_z) \\ \Sigma_g &\approx \Sigma_z \odot \nabla \Psi(\mu_z) \nabla \Psi(\mu_z)^T \end{aligned} \quad (7)$$

Similarly, to reduce computational complexity, the diagonal variance elements are vectorized, and the covariance elements are ignored. The vectorized form of Equation (7) is shown in Equation (8), where σ^2_z represents the vectorized form of the diagonal variance elements. In the vectorized form, the outer product used for constructing the correlation matrix in Equation (7) can be replaced by the squared gradient of the non-linear activation function with respect to the input mean due to the off-diagonal elements are no longer required for the Hadamard product with the input covariance matrix.

$$\begin{aligned} \mu_g &\approx \Psi(\mu_z) \\ \sigma^2_{g_i} &\approx \sigma^2_{z_i} \odot \left(\frac{\partial \Psi}{\partial z_i}(\mu_z) \right)^2 \end{aligned} \quad (8)$$

For the Softmax classification layer at the output of the network $\mu_{\hat{y}}$ and $\sigma^2_{\hat{y}}$ are representative of the variational distribution that can be used to infer the ELBO objective function.

C. Closed Form ELBO

Recalling the objective function for Variational Inference, presented in Equation (4), using the propagated values $\mu_{\hat{y}}$ and $\sigma^2_{\hat{y}}$ of the variational distribution $q(\Omega)$ the ELBO can be written in closed form, as shown in Equation (9). The weighting variable τ is added to control the level of compression toward the prior induced by the KL divergence term in the ELBO. The vectorized form of the closed-form ELBO changes the log determinant of the covariance matrix $\Sigma_{\hat{y}}$ to

$$ELBO = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln(|\Sigma_{\hat{y}}|) - \frac{1}{2}(y-\mu_{\hat{y}})^T \Sigma_{\hat{y}}^{-1}(y-\mu_{\hat{y}}) - \frac{\tau}{2}\sum_{i=1}^{|\Omega|}\left(-1 + \frac{(\mu_{q_{w_i}} - \mu_{p_i})^2}{\sigma^2_{q_{w_i}}} + \ln\left(\frac{\sigma^2_{p_i}}{\sigma^2_{q_{w_i}}}\right) + \frac{\sigma^2_{q_{w_i}}}{\sigma^2_{p_i}}\right) \quad (9)$$

$$ELBO = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\sum_{n=1}^{N}\ln(\sigma^2_{\hat{y}_n}) + \frac{1}{2}\sum_{n=1}^{N}\frac{(y_n - \mu_{\hat{y}_n})^2}{\sigma^2_{\hat{y}_n}} - \frac{\tau}{2}\sum_{i=1}^{|\Omega|}\left(-1 + \frac{(\mu_{q_{w_i}} - \mu_{p_i})^2}{\sigma^2_{q_{w_i}}} - \ln\left(\frac{\sigma^2_{p_i}}{\sigma^2_{q_{w_i}}}\right) - \frac{\sigma^2_{q_{w_i}}}{\sigma^2_{p_i}}\right) \quad (10)$$

the log sum of all the variational distributions, effectively producing the log product of the diagonal variance values from the covariance matrix, equivalent to the determinant when the off-diagonal elements are zero. The same is true for $\Sigma_{\hat{y}}^{-1}$ where the inverse of a diagonal matrix is simply the inverse of each diagonal element. The vectorized closed-form expression of the ELBO is shown in Equation (10) where $|\Omega|$ denotes the cardinality of the set Ω, i.e., the number of weight parameters and N denotes the number of classes or output nodes of the final layer.

III. Continual Learning Methodology

A. Continual Learning

Continual Learning aims to retain performance over multiple training periods. Many approaches have been developed to combat the problem of catastrophic forgetting, separating into three main categories: Regularization, Replay, and Dynamic Architectures [13]. The Dynamic Architectures approach focuses on adding additional neural resources to a network to account for information in new tasks. Replay focuses on preserving certain data samples from previous tasks and "replaying" them during the training of subsequent tasks. Regularization-based approaches constrain the optimization problem to prevent representations of previous data from adapting exclusively to the new data in the network. The method outlined in this paper aligns with Regularization based Continual Learning, where the KL Divergence regularizes the optimization to restrict parameters to remain close to the previous task.

B. Prior Compression

Continually learning with Variational Density Propagation leverages the KL Divergence between the variational posterior and the prior in the ELBO objective function. This regulates learning to prevent network parameters from drifting far from the prior. The model parameters are treated as independent random variables to obtain a simplified version of the KL divergence between the variational posterior and the prior, as shown in Equation 10. This formulation allows each parameter to receive independent gradient updates from the KL term so the new posterior remains close to the previous posterior while receiving updates from the model likelihood. This is an ancillary benefit of assuming all explicitly parameterized random variables are independent.

When learning over multiple tasks, the prior $p_t(\Omega)$ iteratively becomes the posterior of the previous task $q(\Omega_{t-1})$ for task t. The updated prior, $q(\Omega_{t-1})$, contains all information from all previously trained task representations, such that $p_t(\Omega) = q_t(\Omega_{t-1})$. For the first task, $t = 0$, the prior is chosen as standard normal Gaussian for every network parameter. Using a standard normal prior is known to be parameter sparsity-inducing [9]. Using the KL divergence term to constrain network parameters near zero will effectively remove unneeded model parameters from contributing to a model's prediction.This process aligns with the idea of the Minimum Description Length principle [14] in which the best model for a given dataset is the one that results in the minimum total description length of the dataset together with the model, satisfying Occam's Razor principle [4]. By converging to the minimal model complexity on the first task, all subsequent tasks will retain the goal of minimal model complexity via updating the prior distribution to the variation posterior. Thus, a sparsity-inducing prior is effectively applied across all tasks. This process is shown in Algorithm 1.

Algorithm 1 Continual Learning via Prior Compression

Require: $D = (X, y)$;
  The predictive distribution $\hat{y}_m \sim N(\mu_{\hat{y}_m}, \sigma^2_{\hat{y}_m})$;
  The variational distribution $q_\theta(\Omega) \sim \prod_j N(\mu_{w_j}, \sigma^2_{w_j})$;
  The prior distribution $p_0(\Omega_0) \sim \prod N(0,1)$;
  KL divergence weighting factor τ;
  $\tau = \tau_0$
  $q_{\theta_0^*}(\Omega_0) \leftarrow \text{argmax}_\theta[E_{q_\theta(\Omega_0)}[\ln p(y|X,\Omega_0)] - \tau KL_{q_\theta(\Omega_0)}[q_\theta(\Omega_0)||p_0(\Omega_0)]]$
  $\tau = \tau_1$
  for Task $t > 0$ do
    $p_t(\Omega_t) \leftarrow q_{\theta_{t-1}^*}(\Omega_{t-1})$
    $q_{\theta_t^*}(\Omega_t) \leftarrow \text{argmax}_\theta[E_{q_\theta(\Omega_t)}[\ln p(y|X,\Omega_t)] - \tau KL_{q_\theta(\Omega_t)}[q_\theta(\Omega_t)||p_t(\Omega_t)]]$
  end for

C. Experimental Setup

Our Continual Learning framework is evaluated for task incremental learning in which task information is given at test time and the network has separate output layers per task, referred to as a multi-headed network [8]. In this multi-headed architecture, all parameters in layers before the classification layer are shared amongst all tasks. Multi-headed networks promote feature sharing by encouraging the slight differences required between tasks to be captured in the bespoke classification layer [15]. We evaluate task incremental learning on the MNIST and CIFAR10 benchmark datasets with two versions of task groupings: 5-Split, where there are five different tasks of differentiating between two classes, and 2-Split, where there are two different tasks of differentiating between five classes. Our methodology is also evaluated on Permuted MNIST, in which the MNIST dataset

digit's pixels are permuted with ten different functions, one for each task. The network is then tasked with classifying the permuted digits 0-10. For the Split and Permuted MNIST datasets, a fully connected network with a single 1200-node hidden layer is used. For the sequential CIFAR10 datasets, a six-layer convolutional network with three linear layers is used.

1) Hyperparameters: A hyper-prior was used to initialize the variational posterior where values of mean and variance of the weights were randomly selected from $N(0, 0.05)$ and $N(\pi, 0.05)$, respectively. To ensure the positivity in the variance of network parameters, parameter variance values are passed through the soft-plus activation function before actual network operations, shown in Equation 13. For the first task only, the variational prior was set to a sparsity-inducing, the standard normal prior for every parameter, $N(1, 0)$.

$$\sigma_w^2 = \log(1 + \exp(\sigma_\pi^2)) \tag{13}$$

A grid search is performed to determine the most appropriate initialization scheme for the variance of the weights and KL divergence weighting. The sweep for the mean of the initialization distribution for the variance, π, considered integer values from -6 to -18. Network layer biases were initialized with the same scheme. The KL divergence term between the variational posterior and the prior manages the trade-off between updates from error embedded in the model likelihood and the divergence from the prior. A weighting term τ is applied and varied to ensure sufficient representation of each task was learned and retained over multiple tasks. Similar approaches to approximating Bayesian inference in deep learning using Autoencoders [16] use values of $\tau > 1$ to promote sparsity in the learned latent representation of the data. Using $\tau > 0.001$ would not allow the network to learn a sufficiently complex network to represent the data when learning the first task with a sparsity-inducing prior. The sweep for the hyperparameter τ considered values 1e-3 to 1e-6 for the first task and values between 1e-1 and 1e-4 for all subsequent tasks, with increments of powers of ten. Larger values of τ provide more compression toward the previous tasks posterior. All model variations are trained with the Adam optimizer with an initial learning rate of 1e-3. The learning rate is reduced by a factor of 10 when the loss of each task plateaus. Random seeds are held constant throughout hyperparameter sweeps.

D. Performance Measurement

After learning all test sets for t tasks, models are evaluated to demonstrate performance and catastrophic forgetting mitigation over multiple tasks. Performance is gauged through the Average Test Classification Accuracy (ACC) and Backward Transfer (BWT). Average Test Classification Accuracy is an average of all test accuracies on individual tasks after training all tasks. Backward Transfer indicates how much learning new information has affected performance on previous tasks. Backward Transfer values less than zero indicate catastrophic forgetting, while values greater than zero indicate improved performance on previous tasks after training on new information [17]. If Backward Transfer is greater than zero, then training subsequent tasks improved the generalization of previously trained tasks with information from subsequently trained tasks. These metrics are shown mathematically in Equation 14.

$$\begin{aligned} BWT &= \frac{1}{t}\sum_{i=1}^{t} R_{i,t} - R_{i,i} \\ ACC &= \frac{1}{t}\sum_{i=1}^{t} R_{i,t} \end{aligned} \tag{14}$$

IV. Results and Discussion

Continual Learning performance with our Variational Density Propagation framework (VDP PC) is compared to Variational Continual Learning (VCL) [8], which utilizes sampling-based deep variational inference to mitigate catastrophic forgetting via the approximation of Bayesian Inference. Additionally, both approaches are compared to four baselines in which deterministic and VDP frameworks are trained sequentially: a Single-Head deterministic architecture (DET-SH); Fine Tuning (*-FT), where no efforts are made to mitigate catastrophic forgetting; Feature Extraction (*-FE), where shared network parameters are frozen, but bespoke output layers are trained; and Joint Training (*-JT), where later tasks are supplemented with all data from previous tasks. The Average Test Classification Accuracy (ACC) and Backward Transfer (BWT) metrics across both frameworks and all datasets are presented in Table I.

Despite benchmark datasets' simplistic nature, tasks learned sequentially in a single-headed network (DET-SH) result in a complete loss in performance on previous tasks. Multi-headed networks trained sequentially to fine-tune (*-FT) each new task fare significantly better, retaining the majority of performance after training a sequence of tasks. The bespoke output layer, however, carries the majority of the performance improvement, as demonstrated by freezing all shared parameters (*-FE) before training subsequent tasks and only learning the bespoke output layer. Restricting changes in model complexity with our VDP Prior Compression (VDP PC) approach improves Average Test Classification Accuracy and Backward Transfer performance over all tested sequences of tasks. Performance with the VDP Prior Compression approach closely follows the joint training (*-JT) performance for both the deterministic and VDP frameworks and is considered the upper bound on performance for the tested model architecture on each experiment.

Conversely, we show that Variational Continual Learning (VCL), not trained with a core set, does not improve over existing deterministic baseline approaches. Results collected for VCL are slightly improved over what is reported for Split and Permuted MNIST datasets in the original publication. This improvement is expected to result from reducing the depth of the fully-connected network. Less shared parameters results in less interference from one task to the next. Split Cifar10 results were not collected in the original VCL paper and are demonstrated to follow the same general trend as the

TABLE I
TASK INCREMENTAL LEARNING RESULTS

Approach	2-Split MNIST		5-Split MNIST		Permuted MNIST		2-Split CIFAR10		5-Split CIFAR10	
	ACC	BWT	ACC	BWT	ACC	BWT	ACC	BWT	ACC	BWT
DET-SH	50.72%	-48.42%	20.00%	-79.80%	72.81%	-25.63%	46.50%	-42.89%	19.03%	-74.45%
DET-FT	98.64%	-0.60%	99.34%	-0.49%	96.47%	-1.67%	81.96%	-18.04%	77.68%	-15.59%
VDP-FT	98.55%	-0.35%	98.32%	-1.46%	95.91%	-2.47%	78.82%	-8.53%	73.42%	-19.92%
DET-FE	98.96%	0.00%	99.33%	0.00%	96.41%	0.00%	72.78%	-4.24%	78.43%	-0.79%
VDP-FE	97.90%	0.00%	99.40%	0.00%	96.72%	0.00%	79.49%	0.02%	80.63%	-0.13%
VCL	98.04%	-0.86%	99.08%	-0.50%	88.80%	-7.90%	64.39%	-10.93%	76.82%	-8.96%
VDP PC	99.24%	-0.01%	99.80%	-0.06%	97.71%	-0.14%	88.81%	-1.08%	83.23%	-0.62%
DET-JT	99.38%	-0.01%	99.87%	0.00%	98.33%	-0.10%	86.51%	+0.63%	93.11%	+0.16%
VDP-JT	99.34%	+0.18%	99.73%	-0.08%	98.15%	-0.17%	87.81%	+1.84%	94.04%	+0.19%

MNIST results, performing slightly under fine-tuning performance, demonstrating no additional benefit to catastrophic forgetting mitigation.

V. CONCLUSION

In this work, the Variational Continual Learning framework is improved by removing the requirement for Monte Carlo sampling of the variational posterior of the model parameters. Network operations are replaced with operations of random variables providing a quicker and less noisy estimate of model parameters utilizing a completely closed-form Evidence Lower Bound objective. The inherent compression within the ELBO approximates the Minimum Description Length Principle by penalizing additional model complexity over multiple tasks. Our approach demonstrates catastrophic forgetting mitigation in the task incremental learning setting for a multi-headed network on common Continual Learning benchmark datasets and demonstrates improvement over sampling-based approaches. We leverage the KL regularization weighting term to control the amount of change in model complexity induced by each task. Overall, improved Average Test Classification Accuracy and Backward Transfer metrics are achieved.

REFERENCES

[1] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in non-stationary environments: A survey," IEEE Computational Intelligence Magazine, vol. 10, no. 4, pp. 12–25, 2015.

[2] R. M. French, "Catastrophic forgetting in connectionist networks," Trends in Cognitive Sciences, vol. 3, no. 4, pp. 128–135, 1999.

[3] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," vol. 24 of Psychology of Learning and Motivation, pp. 109–165, Academic Press, 1989.

[4] X. He and M. Lin, "Continual Learning from the Perspective of Compression," arXiv e-prints, p. arXiv:2006.15078, June 2020.

[5] P. D. Grünwald, The Minimum Description Length Principle. Adaptive Computation and Machine Learning Ser., Cambridge: MIT Press, 1st ed. ed., 2007.

[6] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," arXiv e-prints, p. arXiv:1505.05424, May 2015.

[7] Y. Gal and Z. Ghahramani, "Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference," arXiv e-prints, p. arXiv:1506.02158, June 2015.

[8] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational Continual Learning," arXiv e-prints, p. arXiv:1710.10628, Oct. 2017.

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv e-prints, p. arXiv:1312.6114, Dec. 2013.

[10] D. Dera, N. C. Bouaynaya, G. Rasool, R. Shterenberg, and H. M. Fathallah-Shaykh, "Premium-cnn: Propagating uncertainty towards robust convolutional neural networks," IEEE Transactions on Signal Processing, vol. 69, pp. 4669–4684, 2021.

[11] D. Dera, G. Rasool, and N. Bouaynaya, "Extended variational inference for propagating uncertainty in convolutional neural networks," in 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6, 2019.

[12] A. Papoulis, Probability, random variables, and stochastic processes. McGraw-Hill series in electrical engineering. Communications and signal processing, New York: McGraw-Hill, fourth edition. ed., 2002.

[13] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," Neural networks, vol. 113, pp. 54–71, 2019.

[14] G. E. Hinton and R. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in Advances in Neural Information Processing Systems (J. Cowan, G. Tesauro, and J. Alspector, eds.), vol. 6, Morgan-Kaufmann, 1993.

[15] B. Bakker and T. Heskes, "Task clustering and gating for bayesian multitask learning," J. Mach. Learn. Res., vol. 4, p. 83–99, dec 2003.

[16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in International Conference on Learning Representations, 2017.

[17] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided Continual Learning with Bayesian Neural Networks," arXiv e-prints, p. arXiv:1906.02425, June 2019.